Magnet Project API Documentation

Route Authentication & Role Protection Summary

Module	Route/Action	Public	Auth Required	Role(s) Required
Products	GET /api/products	No	Yes	Any authenticated user
	GET /api/products/:id	No	Yes	Any authenticated user
	POST /api/products/addProductsByBusiness	No	Yes	business
	POST /api/products/addProductsByMagnet_employee	No	Yes	magnet_employee, admin
	PUT /api/products/:id	No	Yes	business (own), admin, magnet_employee
	DELETE /api/products/:id	No	Yes	business (own), admin, magnet_employee
	PUT /api/products/:id/approve	No	Yes	admin, magnet_employee
	PUT /api/products/:id/decline	No	Yes	admin, magnet_employee
Categories	GET /api/categories	Yes	No	None
	POST /api/categories	No	Yes	business, admin, magnet_employee
	PUT /api/categories/:id	No	Yes	business (own), admin, magnet_employee
	DELETE /api/categories/:id	No	Yes	business (own), admin, magnet_employee
Orders	POST /api/orders	No	Yes	customer
	GET /api/orders/my	No	Yes	customer
	GET /api/orders/:id	No	Yes	admin, magnet_employee, customer (own)

Module	Route/Action	Public	Auth Required	Role(s) Required
	GET /api/orders	No	Yes	admin, magnet_employee
Reviews	POST /api/reviews/products/:id/reviews	No	Yes	customer
	GET /api/reviews/products/:id/reviews	Yes	No	None
	DELETE /api/reviews/:id	No	Yes	admin, magnet_employee
Wishlist	GET /api/wishlist	No	Yes	customer
	POST /api/wishlist	No	Yes	customer
	DELETE /api/wishlist/:productId	No	Yes	customer
Address	GET /api/addresses	No	Yes	customer
	POST /api/addresses	No	Yes	customer
	PUT /api/addresses/:id	No	Yes	customer
	DELETE /api/addresses/:id	No	Yes	customer
User	GET /api/user/profile	No	Yes	customer
	PUT /api/user/profile	No	Yes	customer
	GET /api/user/business-requests	No	Yes	admin, magnet_employee
	POST /api/user/business-approval	No	Yes	admin, magnet_employee
	GET /api/user/business/:businessId	No	Yes	admin, magnet_employee
	GET /api/user/business-profile	No	Yes	business, admin, magnet_employee
Auth	Registration/Login/OTP	Yes	No	None
	Forgot Password	No	Yes	Any authenticated user

Notes

- **Auth Required**: Requires a valid JWT token in the `Authorization` header.
- **Role(s) Required**: Requires the user to have one of the listed roles. If "(own)" is specified, the user must own the resource (e.g., their own product or order).
- **Public**: Anyone can access, no authentication required.
- **All**: All HTTP methods (GET, POST, PUT, DELETE) for that endpoint.

- **Business (if owns product in order)**: Business user can access if any product in the order is owned by them (see order tracking section).

Message Format

All API responses return bilingual messages in both English and Arabic for every endpoint, including all errors and successes. The system uses a centralized messages.js file to ensure consistency across all modules (auth, user, product, order, address, category, review, wishlist, etc.).

Example:

```
{
  "status": "success",
  "message": {
    "en": "User registered successfully",
    "ar": "تم تسجيل المستخدم بنجاح"
  },
  "data": { ... }
}
```

Verification Rules

Phone Number Verification

- **If phone number is provided**: Only the phone is verified (`isPhoneVerified = true`, `isEmailVerified = false`)
- **If no phone number is provided**: Only the email is verified (`isEmailVerified = true`, `isPhoneVerified = false`)

This applies to both customer and business registration.

Validation Rules

Email Validation

- **Format**: Must match email format: `^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$`
- **Required**: Email is required for all user registrations
- **Unique**: Email must be unique across all users

Phone Validation

- **Format**: Must match phone format: `^\+?[1-9]\d{1,14}$`
- **Optional**: Phone number is optional for customer registration
- **Unique**: If provided, phone must be unique across all users
- **Saudi Numbers**: Saudi phone numbers are automatically verified

Password Requirements

- **Minimum Length**: 8 characters
- **Complexity**: Must contain at least:
  - One uppercase letter (A-Z)
  - One lowercase letter (a-z)
  - One number (0-9)
- **Pattern**: `^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)`

Name Validation

- **First Name**: 2-50 characters, required
- **Last Name**: 2-50 characters, required
- **Trim**: Leading and trailing whitespace is automatically removed

Business Registration Validation

- **CR Number**: 1-50 characters, required
- **VAT Number**: 1-50 characters, required
- **Company Name**: 2-100 characters, required
- **Company Type**: 2-100 characters, required
- **Country**: 2-50 characters, required
- **City**: 2-50 characters, required
- **District**: 2-50 characters, required
- **Street Name**: 2-100 characters, required

Product Validation

- **Category**: Required bilingual object `{ en: "English Category", ar: "Arabic Category" }`
- **Name**: Required bilingual object `{ en: "English Name", ar: "Arabic Name" }`
- **Description**: Optional bilingual object `{ en: "English Description", ar: "Arabic Description" }`
- **Unit**: Optional bilingual object `{ en: "English Unit", ar: "Arabic Unit" }`
- **Custom Fields**: Array of objects, minimum 3, maximum 10
  - Each object must have `key` and `value` properties
  - Both key and value must be bilingual objects: `{ en: "English", ar: "Arabic" }`
- **Code**: Optional, auto-generated if not provided
- **Images**: Optional array of strings (URLs)
- **Attachments**: Optional array of product IDs
- **Min Order**: Optional number
- **Price Per Unit**: Optional string
- **Stock**: Optional number

Address Validation

- **Address Line 1**: Required string
- **Address Line 2**: Optional string
- **City**: Required string
- **State**: Required string

- **Postal Code**: Required string
- **Country**: Required string

Category Validation

- **Name**: Required bilingual object `{ en: "English", ar: "Arabic" }`
- **Description**: Optional bilingual object `{ en: "English", ar: "Arabic" }`

Review Validation

- **Rating**: Required number, min: 1, max: 5
- **Comment**: Required string

Order Validation

- **Customer**: Required user ID
- **Items**: Required array of order items
  - Each item must have `product` (product ID) and `quantity` (number)
- **Shipping Address**: Required address ID
- **Status**: Optional, defaults to 'pending', enum: 'pending', 'confirmed', 'shipped', 'delivered', 'cancelled' (English values only, backend converts to bilingual)
  - **Valid English Statuses**: `['pending', 'confirmed', 'shipped', 'delivered', 'cancelled']`
  - **Valid Arabic Statuses**: `['قيد الانتظار', 'مؤكد', 'تم الشحن', 'تم التوصيل', 'ملغي']`
  - **Validation**: Backend validates status using `Order.isValidStatus()` method
  - **Conversion**: Backend converts English status to bilingual using `Order.convertStatusToBilingual()` method

---

Auth Routes

1. Register User

- **POST** `/api/auth/register`
- **Description:** Register a new customer user with automatic verification based on phone number presence.
- **Body:**
  - `firstname` (string, required)
  - `lastname` (string, required)
  - `email` (string, required)
  - `phone` (string, optional)
  - `password` (string, required)
  - `country` (string, required)
  - `language` (string, optional, default: 'en')
- **Verification Logic:**
  - If `phone` is provided: `isPhoneVerified = true`, `isEmailVerified = false`
  - If `phone` is not provided: `isEmailVerified = true`, `isPhoneVerified = false`
- **Response:**

- `201 Created`: User info + JWT token with verification status

---

2. Register Business

- **POST** `/api/auth/business-register`
- **Description:** Register a new business user (requires approval) with automatic verification based on phone number presence.
- **Body:**
  - `firstname`, `lastname`, `email`, `password`, `crNumber`, `vatNumber`, `companyName`, `companyType`, `country`, `city`, `district`, `streetName`, `phone` (all required)
- **Verification Logic:**
  - If `phone` is provided: `isPhoneVerified = true`, `isEmailVerified = false`
  - If `phone` is not provided: `isEmailVerified = true`, `isPhoneVerified = false`
- **Response:**
  - `201 Created`: Business info (under review) with verification status

---

3. Send Email OTP

- **POST** `/api/auth/send-email-otp`
- **Description:** Send an OTP to a new email (fails if email already exists).
- **Body:**
  - `email` (string, required)
- **Response:**
  - `200 OK`: Success message (bilingual)

---

4. Send Phone OTP

- **POST** `/api/auth/send-phone-otp`
- **Description:** Send an OTP to a new phone (fails if phone already exists).
- **Body:**
  - `phone` (string, required)
- **Response:**
  - `200 OK`: Success message (bilingual)

---

5. Confirm OTP

- **POST** `/api/auth/confirm-otp`
- **Description:** Confirm an OTP for any email or phone identifier (works for both registered and non-registered users).
- **Body:**
  - `identifier` (string, required)
  - `otp` (string, required)
- **Response:**
  - `200 OK`: OTP verified successfully (bilingual message)

---

6. Confirm Login OTP

- **POST** `/api/auth/confirm-login-otp`
- **Description:** Confirm an OTP for registered users and return user data with JWT token (like login API).
- **Body:**
  - `identifier` (string, required)
  - `otp` (string, required)
- **Response:**
  - `200 OK`: User info + JWT token (bilingual message)
  - `404 Not Found`: User not found (bilingual message)

---

7. Login

- **POST** `/api/auth/login`
- **Description:** Login with email/phone and password.
- **Body:**
  - `identifier` (email or phone, required)
  - `password` (string, required)
- **Response:**
  - `200 OK`: User info + JWT token (bilingual message)

---

8. Login with OTP

- **POST** `/api/auth/login-with-otp`
- **Description:** Request an OTP for login (email or phone).
- **Body:**
  - `identifier` (email or phone, required)
- **Response:**
  - `200 OK`: OTP sent (bilingual message)

---

9. Forgot Password

- **POST** `/api/auth/forgot-password`
- **Description:** Change password for authenticated user.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `newPassword` (string, required)
- **Response:**
  - `200 OK`: Password updated (bilingual message)

---

10. Create Admin User

- **POST** `/api/auth/create-admin`
- **Description:** Create a new admin user. Only accessible by authenticated admin users.
- **Headers:** `Authorization: Bearer <token>`

- **Body:**
  - `firstname` (string, required)
  - `lastname` (string, required)
  - `email` (string, required)
  - `phone` (string, optional)
  - `password` (string, required)
  - `country` (string, required)
  - `language` (string, optional, default: 'en')
- **Response:**
  - `201 Created`: Admin user info (bilingual message)
  - `403 Forbidden`: Insufficient permissions (bilingual message)
  - `400 Bad Request`: Email/phone already registered (bilingual message)

---

11. Create Magnet Employee User

- **POST** `/api/auth/create-magnet-employee`
- **Description:** Create a new magnet employee user. Only accessible by authenticated admin users.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `firstname` (string, required)
  - `lastname` (string, required)
  - `email` (string, required)
  - `phone` (string, optional)
  - `password` (string, required)
  - `country` (string, required)
  - `language` (string, optional, default: 'en')
- **Response:**
  - `201 Created`: Magnet employee user info (bilingual message)
  - `403 Forbidden`: Insufficient permissions (bilingual message)
  - `400 Bad Request`: Email/phone already registered (bilingual message)

Product Routes

Language Support

Products support bilingual content (Arabic and English). You can:

- **Get products in specific language**: Add `?lang=en` or `?lang=ar` to the URL
- **Get products in user's preferred language**: The system uses the user's language preference
- **Get full bilingual data**: Use `?lang=both` to get both languages

1. Get Products

- **GET** `/api/products`
- **GET** `/api/products?lang=en` (English only)
- **GET** `/api/products?lang=ar` (Arabic only)
- **GET** `/api/products?lang=both` (Both languages)

- **Description:** Get all approved products (requires authentication). Admin/magnet_employee can see all products.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: List of products
- **Product Object:**
  - `code` (string, auto-generated if not provided, e.g. "A001")
  - `category` (object, required, bilingual: `{ en: "English Category", ar: "Arabic Category" }`)
  - `name` (object, required, bilingual: `{ en: "English Name", ar: "Arabic Name" }`)
  - `images` (array of strings)
  - `description` (object, optional, bilingual: `{ en: "English Description", ar: "Arabic Description" }`)
  - `attachments` (array of product IDs, references to other products)
  - `unit` (object, optional, bilingual: `{ en: "English Unit", ar: "Arabic Unit" }`)
  - `minOrder` (number)
  - `pricePerUnit` (string)
  - `stock` (number)
  - `customFields` (array of objects, min 3, max 10, each: `{ key: { en: "English Key", ar: "Arabic Key" }, value: { en: "English Value", ar: "Arabic Value" } }`)
  - `status` (string: 'pending', 'approved', 'declined')
  - `owner` (user id, required)
  - `approvedBy` (user id, admin/employee who approved)
  - `createdAt` (date)

1a. Get Product by ID

- **GET** `/api/products/:id`
- **GET** `/api/products/:id?lang=en` (English only)
- **GET** `/api/products/:id?lang=ar` (Arabic only)
- **GET** `/api/products/:id?lang=both` (Both languages)
- **Description:** Get a single product by its ID. Requires authentication. Admin, business, and magnet_employee can access any product (including pending/declined).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Product object (see above)
  - `404 Not Found`: If product does not exist
  - `403 Forbidden`: If trying to access a non-approved product without proper role

---

2. Add Product (Business)

- **POST** `/api/products/addProductsByBusiness`
- **Description:** Business user adds a new product (pending approval). Product code is auto-generated if not provided.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**

  - `category` (object, required, bilingual: `{ en: "English Category", ar: "Arabic Category" }`)
  - `name` (object, required, bilingual: `{ en: "English Name", ar: "Arabic Name" }`)
  - `images` (array of strings, optional)
  - `attachments` (array of product IDs, references to other products, optional)
  - `description` (object, optional, bilingual: `{ en: "English Description", ar: "Arabic Description" }`)
  - `unit` (object, optional, bilingual: `{ en: "English Unit", ar: "Arabic Unit" }`)
  - `minOrder`, `pricePerUnit`, `stock` (all optional)
  - `customFields` (array of objects, min 3, max 10, each: `{ key: { en: "English Key", ar: "Arabic Key" }, value: { en: "English Value", ar: "Arabic Value" } }`, required)
- **Response:**
  - `201 Created`: Product info (pending approval) with bilingual message

---

3. Add Product (Magnet Employee)

- **POST** `/api/products/addProductsByMagnet_employee`
- **Description:** Magnet employee adds a new product (approved immediately). Product code is auto-generated if not provided.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `category` (object, required, bilingual: `{ en: "English Category", ar: "Arabic Category" }`)
  - `name` (object, required, bilingual: `{ en: "English Name", ar: "Arabic Name" }`)
  - `images` (array of strings, optional)
  - `attachments` (array of product IDs, references to other products, optional)
  - `description` (object, optional, bilingual: `{ en: "English Description", ar: "Arabic Description" }`)
  - `unit` (object, optional, bilingual: `{ en: "English Unit", ar: "Arabic Unit" }`)
  - `minOrder`, `pricePerUnit`, `stock` (all optional)
  - `customFields` (array of objects, min 3, max 10, each: `{ key: { en: "English Key", ar: "Arabic Key" }, value: { en: "English Value", ar: "Arabic Value" } }`, required)
  - `owner` (user id, required)
- **Response:**
  - `201 Created`: Product info (approved) with bilingual message

---

4. Update Product

- **PUT** `/api/products/:id`
- **Description:** Business user updates their own product (pending approval). Admin/magnet_employee can update and approve/decline.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - Any product field (see above)
  - `status` (optional, only admin/magnet_employee can set to 'approved' or 'declined')
- **Response:**

- `200 OK`: Updated product info with bilingual message

---

5. Delete Product

- **DELETE** `/api/products/:id`
- **Description:** Business user deletes their own product, or admin/employee deletes any product.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Product deleted with bilingual message

---

6. Approve Product

- **PUT** `/api/products/:id/approve`
- **Description:** Admin/magnet_employee approves a product.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Product approved with bilingual message

---

7. Decline Product

- **PUT** `/api/products/:id/decline`
- **Description:** Admin/magnet_employee declines a product.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Product declined with bilingual message

Real-Time Order Tracking (WebSocket)

Overview

- The API supports real-time order status updates using WebSocket (Socket.IO).
- Clients can subscribe to updates for specific orders and receive instant notifications when the order status changes (e.g., confirmed, shipped, delivered).

How It Works

1. **Client authenticates with a JWT token when connecting to the WebSocket.**
2. **Client joins a room for a specific order using the order ID.**
3. **When the order status changes, the server emits an `orderStatusUpdate` event to all clients in that room.**

WebSocket Connection (Socket.IO)

- **URL:** `ws://<your-server>:5000` (or `wss://` for HTTPS)
- **Library:** socket.io-client

Client Example (JavaScript/React):

```
import { io } from 'socket.io-client';
const socket = io('http://localhost:5000', {
  auth: { token: '<JWT_TOKEN>' }
});

// Join the order room
socket.emit('joinOrderRoom', '<ORDER_ID>');

// Listen for updates
socket.on('orderStatusUpdate', (data) => {
  // data: { orderId, status, statusLog, updatedAt }
  console.log('Order update:', data);
});
```

Order Room Access Rules

- **Admin & magnet_employee:** Can track any order.
- **Business:** Can track orders that include products they own.
- **Customer:** Can track only their own orders.
- Unauthorized attempts to join order rooms will be ignored.

Server-Side Security

- The server authenticates each socket connection using the JWT token.
- The server enforces the above access rules for joining order rooms.

Order Status Update Event

- **Event:** `orderStatusUpdate`
- **Payload:**

```
{
  "orderId": "...",
  "status": "shipped",
  "statusLog": [
    { "status": "pending", "timestamp": "..." },
    { "status": "confirmed", "timestamp": "..." },
    { "status": "shipped", "timestamp": "..." }
  ],
  "updatedAt": "2024-07-10T12:00:00.000Z"
}
```

Security Notes

- Clients must provide a valid JWT token when connecting.

- The server will only allow joining order rooms for orders the user is authorized to track (see access rules above).
- Unauthorized attempts to join rooms will be ignored.

Additional Utility Features

Email and SMS Utilities

The API includes comprehensive email and SMS functionality for user verification and notifications:

Email Features

- **Email OTP Generation**: Automatic generation of 6-digit verification codes
- **Email Templates**: Pre-formatted HTML email templates for various notifications
- **Business Notifications**: Automatic email notifications for business registration status changes
- **Verification Emails**: Email verification for user registration

SMS Features

- **SMS OTP Generation**: Automatic generation of 6-digit SMS verification codes
- **Saudi Number Support**: Special handling for Saudi phone numbers (+966)
- **SMS Templates**: Pre-formatted SMS messages for various notifications
- **Phone Verification**: SMS verification for user registration

Product Code Generation

The API includes an automatic product code generation system:

Features

- **Auto-Generation**: If no product code is provided, the system automatically generates a unique code
- **Format**: Codes follow the pattern "A001", "A002", etc.
- **Uniqueness**: Each generated code is guaranteed to be unique
- **Custom Codes**: Users can provide their own custom codes if desired

User Model Features

The user model includes several advanced features:

Favorites System

- **Add to Favorites**: Users can add products to their favorites list
- **Remove from Favorites**: Users can remove products from their favorites
- **View Favorites**: Users can view their complete favorites list

Cart System

- **Add to Cart**: Users can add products to their shopping cart
- **Cart Management**: Users can update quantities and remove items

- **Cart Persistence**: Cart data is stored with the user account

Business Approval Workflow

- **Registration Review**: Business registrations require admin/employee approval
- **Status Tracking**: Business approval status is tracked (pending, approved, rejected)
- **Notification System**: Automatic notifications for approval status changes
- **Rejection Reasons**: Admins can provide reasons for business registration rejections

Verification Status Tracking

- **Email Verification**: Track whether user's email is verified
- **Phone Verification**: Track whether user's phone is verified
- **Automatic Verification**: Saudi phone numbers are automatically verified
- **Manual Verification**: Other verification methods available

Bilingual Message System

The API includes a comprehensive bilingual message system:

Features

- **Centralized Messages**: All messages are stored in a centralized `messages.js` file
- **Bilingual Support**: All messages are available in both English and Arabic
- **Consistent Format**: All API responses follow the same bilingual format
- **Error Messages**: Comprehensive error messages in both languages
- **Success Messages**: Success confirmations in both languages

Message Format

```
{
  "status": "success",
  "message": {
    "en": "Operation completed successfully",
    "ar": "تم إكمال العملية بنجاح"
  },
  "data": { ... }
}
```

Authentication and Authorization

The API includes a robust authentication and authorization system:

JWT Authentication

- **Token-Based**: Uses JWT tokens for authentication
- **Expiration**: Tokens expire after 7 days
- **Refresh**: Tokens can be refreshed

- **Secure**: Uses environment variables for secret keys

Role-Based Access Control

- **Admin**: Full system access
- **Magnet Employee**: Limited admin access
- **Business**: Business-specific features
- **Customer**: Customer-specific features

Middleware System

- **Auth Middleware**: Verifies JWT tokens
- **Role Middleware**: Enforces role-based access
- **Validation Middleware**: Validates request data
- **Error Handling**: Comprehensive error handling

Category Routes

Language Support

Categories support bilingual content (Arabic and English). You can:

- **Get categories in specific language**: Add `?lang=en` or `?lang=ar` to the URL
- **Get categories in user's preferred language**: The system uses the user's language preference
- **Get full bilingual data**: Use `?lang=both` to get both languages

Category Object

- `name` (object, required, bilingual: `{ en: "English Name", ar: "Arabic Name" }`)
- `description` (object, optional, bilingual: `{ en: "English Description", ar: "Arabic Description" }`)
- `createdBy` (ObjectId, reference to User)
- `createdAt` (date)
- `updatedAt` (date)

1. Get Categories

- **GET** `/api/categories`
- **GET** `/api/categories?lang=en` (English only)
- **GET** `/api/categories?lang=ar` (Arabic only)
- **GET** `/api/categories?lang=both` (Both languages)
- **Description:** Get all categories (public).
- **Response:**
  - `200 OK`: List of categories

2. Create Category

- **POST** `/api/categories`
- **Description:** Create a new category (business, admin, or magnet_employee only).

- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `name` (object, required, bilingual: `{ en: "English Name", ar: "Arabic Name" }`)
  - `description` (object, optional, bilingual: `{ en: "English Description", ar: "Arabic Description" }`)
- **Response:**
  - `201 Created`: Category info (bilingual message)

3. Update Category

- **PUT** `/api/categories/:id`
- **Description:** Update a category (business, admin, or magnet_employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `name` (object, optional, bilingual: `{ en: "English Name", ar: "Arabic Name" }`)
  - `description` (object, optional, bilingual: `{ en: "English Description", ar: "Arabic Description" }`)
- **Response:**
  - `200 OK`: Updated category info (bilingual message)

4. Delete Category

- **DELETE** `/api/categories/:id`
- **Description:** Delete a category (business, admin, or magnet_employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Category deleted (bilingual message)

Order Routes

Language Support

Orders support bilingual content (Arabic and English). You can:

- **Get orders in specific language**: Add `?lang=en` or `?lang=ar` to the URL
- **Get orders in user's preferred language**: The system uses the user's language preference
- **Get full bilingual data**: Use `?lang=both` to get both languages

Order Object

- `customer` (ObjectId, required, reference to User)
- `items` (array of objects, required)
  - Each item: `{ product: ObjectId, quantity: Number }`
- `shippingAddress` (ObjectId, reference to Address)
- `status` (object, bilingual: `{ en: "pending", ar: "قيد الانتظار" }`)
- `statusLog` (array of objects)
  - Each log: `{ status: { en: "pending", ar: "قيد الانتظار" }, timestamp: Date }`
- `createdAt` (date)

- `updatedAt` (date)

Note: Order status is stored as bilingual objects but returned as localized strings based on the `lang` parameter:

- `pending` → `{ en: "pending", ar: "قيد الانتظار" }`
- `confirmed` → `{ en: "confirmed", ar: "مؤكد" }`
- `shipped` → `{ en: "shipped", ar: "تم الشحن" }`
- `delivered` → `{ en: "delivered", ar: "تم التوصيل" }`
- `cancelled` → `{ en: "cancelled", ar: "ملغي" }`

1. Create Order

- **POST** `/api/orders`
- **Description:** Create a new order (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - Order details (see order model)
- **Response:**
  - `201 Created`: Order info (bilingual message)

2. Get My Orders

- **GET** `/api/orders/my`
- **GET** `/api/orders/my?lang=en` (English only)
- **GET** `/api/orders/my?lang=ar` (Arabic only)
- **GET** `/api/orders/my?lang=both` (Both languages)
- **Description:** Get all orders for the authenticated customer.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: List of orders (bilingual message)

3. Get Order by ID

- **GET** `/api/orders/:id`
- **GET** `/api/orders/:id?lang=en` (English only)
- **GET** `/api/orders/:id?lang=ar` (Arabic only)
- **GET** `/api/orders/:id?lang=both` (Both languages)
- **Description:** Get order by ID (admin, magnet_employee, or owner).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Order info (bilingual message)

4. Get All Orders

- **GET** `/api/orders`
- **GET** `/api/orders?lang=en` (English only)
- **GET** `/api/orders?lang=ar` (Arabic only)
- **GET** `/api/orders?lang=both` (Both languages)

- **Description:** Get all orders (admin, magnet_employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: List of orders (bilingual message)

5. Update Order Status

- **PUT** `/api/orders/:id/status`
- **Description:** Update order status (admin, magnet_employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `status` (string, required, enum: 'pending', 'confirmed', 'shipped', 'delivered', 'cancelled')
- **Response:**
  - `200 OK`: Updated order info (bilingual message)

6. Get Status Options

- **GET** `/api/orders/status-options`
- **GET** `/api/orders/status-options?lang=en` (English only)
- **GET** `/api/orders/status-options?lang=ar` (Arabic only)
- **GET** `/api/orders/status-options?lang=both` (Both languages)
- **Description:** Get available order status options (public).
- **Response:**
  - `200 OK`: Status options in requested language
  - **Example Response:**


```
{
  "status": "success",
  "data": {
    "statusOptions": {
      "pending": "قيد الانتظار",
      "confirmed": "مؤكد",
      "shipped": "تم الشحن",
      "delivered": "تم التوصيل",
      "cancelled": "ملغي"
    },
    "statusEnums": {
      "en": ["pending", "confirmed", "shipped", "delivered",
"cancelled"],
      "ar": ["","تم التوصيل" ,"تم الشحن" ,"مؤكد" ,"قيد الانتظار",
ملغي""]
    },
    "validEnglishStatuses": ["pending", "confirmed", "shipped",
"delivered", "cancelled"]
  }
}
```

Address Routes

Language Support

Addresses no longer support bilingual content. All address fields are stored as simple strings.

Address Object

- `user` (ObjectId, required, reference to User)
- `addressLine1` (string, required)
- `addressLine2` (string, optional)
- `city` (string, required)
- `state` (string, required)
- `postalCode` (string, required)
- `country` (string, required)
- `createdAt` (date)
- `updatedAt` (date)

1. Get Addresses

- **GET** `/api/addresses`
- **Description:** Get all addresses for the authenticated customer.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: List of addresses (bilingual message)

2. Add Address

- **POST** `/api/addresses`
- **Description:** Add a new address (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `addressLine1` (string, required)
  - `addressLine2` (string, optional)
  - `city` (string, required)
  - `state` (string, required)
  - `postalCode` (string, required)
  - `country` (string, required)
- **Response:**
  - `201 Created`: Address info (bilingual message)

3. Update Address

- **PUT** `/api/addresses/:id`
- **Description:** Update an address (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - Any address field (see above)

- **Response:**
  - `200 OK`: Updated address info (bilingual message)

4. Delete Address

- **DELETE** `/api/addresses/:id`
- **Description:** Delete an address (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Address deleted (bilingual message)

Wishlist Routes

1. Get Wishlist

- **GET** `/api/wishlist`
- **Description:** Get the authenticated customer's wishlist.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: List of wishlist items (bilingual message)

2. Add to Wishlist

- **POST** `/api/wishlist`
- **Description:** Add a product to the authenticated customer's wishlist.
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `productId` (string, required)
- **Response:**
  - `200 OK`: Wishlist item info (bilingual message)

3. Remove from Wishlist

- **DELETE** `/api/wishlist/:productId`
- **Description:** Remove a product from the authenticated customer's wishlist.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Wishlist item removed (bilingual message)

Review Routes

Language Support

Reviews no longer support bilingual content. All review comments are stored as simple strings.

Review Object

- `product` (ObjectId, required, reference to Product)

- `user` (ObjectId, required, reference to User)
- `rating` (number, required, min: 1, max: 5)
- `comment` (string, required)
- `createdAt` (date)

1. Add Review

- **POST** `/api/reviews/products/:id/reviews`
- **Description:** Add a review to a product (customer only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `rating` (number, required, min: 1, max: 5)
  - `comment` (string, required)
- **Response:**
  - `201 Created`: Review info (bilingual message)

2. Get Product Reviews

- **GET** `/api/reviews/products/:id/reviews`
- **Description:** Get all reviews for a product (public).
- **Response:**
  - `200 OK`: List of reviews (bilingual message)

3. Delete Review

- **DELETE** `/api/reviews/:id`
- **Description:** Delete a review (admin or magnet_employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Review deleted (bilingual message)

User Routes

1. Get User Profile

- **GET** `/api/user/profile`
- **Description:** Get the authenticated user's profile.
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: User profile info (bilingual message)

2. Update User Profile

- **PUT** `/api/user/profile`
- **Description:** Update the authenticated user's profile.
- **Headers:** `Authorization: Bearer <token>`

- **Body:**
  - `firstname` (string, optional)
  - `lastname` (string, optional)
  - `email` (string, optional)
  - `phone` (string, optional)
  - `country` (string, optional)
  - `language` (string, optional)
- **Response:**
  - `200 OK`: Updated user profile (bilingual message)

3. Get Business Requests

- **GET** `/api/user/business-requests`
- **Description:** Get all business registration requests (admin/employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: List of business requests (bilingual message)

4. Business Approval

- **POST** `/api/user/business-approval`
- **Description:** Approve or reject business registration (admin/employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Body:**
  - `businessId` (string, required)
  - `action` (string, required, enum: 'approve', 'reject')
  - `reason` (string, optional, required if action is 'reject')
- **Response:**
  - `200 OK`: Business approval status updated (bilingual message)

5. Get Business Details

- **GET** `/api/user/business/:businessId`
- **Description:** Get business details by ID (admin/employee only).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Business details (bilingual message)

6. Get Business Profile

- **GET** `/api/user/business-profile`
- **Description:** Get business profile (business users only).
- **Headers:** `Authorization: Bearer <token>`
- **Response:**
  - `200 OK`: Business profile info (bilingual message)